

Analisis Unjuk Kerja *Flow Control* pada *Network on Chip* dalam Beberapa Kondisi Jaringan

M Hizrian Hizburrahman, Istas Pratomo, dan Djoko Suprajitno
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: istaspra@ee.its.ac.id , djokosr@ee.its.ac.id

Abstrak—Semakin berkembangnya teknologi, ukuran dari perangkat keras yang digunakan pada perangkat elektronik semakin berukuran kecil dan menggunakan teknologi *multiprocessor*. Sehingga digunakan teknik *System on Chip* untuk mengatur hubungan antar prosesor - prosesor yang ada. *Network on Chip* ialah teknik yang digunakan di *System on Chip* sebagai pengganti *shared bus* dan *direct point – to – point*. Pada *Network on Chip* terdapat parameter desain dan parameter performansi jaringan. Penentuan parameter desain dan perkembangan dari jaringan dapat menimbulkan permasalahan pada jaringan seperti *congestion* dan saturasi yang menyebabkan paket hilang. *Congestion* dapat diatasi dengan menggunakan *flow control* yang tepat. Pada penelitian ini dilakukan analisis terhadap tiga teknik *flow control* yaitu Stall / Go , Ack / Nack serta Dynamic Multi Level yang diterapkan pada dua model jaringan. Model jaringan yang pertama untuk mengamati pengaruh *flow control* terhadap saturasi jaringan dan model kedua untuk mengamati pengaruh *flow control* terhadap perubahan parameter desain dan mendapatkan teknik *flow control* yang paling optimal dan pengaruh perubahan parameter desain terhadap parameter performansi jaringan. Dari hasil penelitian, jaringan yang menggunakan *flow control* tidak mengalami saturasi. Dimana *flow control* Stall / Go merupakan *flow control* terbaik dalam meningkatkan *throughput* sebesar 21.08% , 65.33%, 151% dan 13.37% , menurunkan *delay* sebesar 407.85, 606.03, 1631.95, 322.59 cycles, menurunkan penggunaan daya sebesar 68.67%, 61.33%, 49.93%, 68.22% untuk masing – masing perubahan ukuran jaringan, perubahan *packet injection rate*, perubahan ukuran paket dan perubahan ukuran *buffer*.

Kata Kunci—*Congestion* , *Flow Control* , *Network on Chip*, *System on Chip*

I. PENDAHULUAN

SEMAKIN berkembangnya teknologi, ukuran dari perangkat keras yang digunakan pada perangkat elektronik berubah menuju ke ukuran yang semakin kecil. Dimana perangkat elektronik menggunakan teknologi *multiprocessor* dengan ukuran yang sangat kecil. Teknik *System on Chip* digunakan untuk mengatur hubungan antar processor – processor yang ada. Processor – processor yang digunakan pada SoC disebut sebagai *Processing Element* (PE). Dimana setiap PE dapat saling berkomunikasi satu sama lain. Teknik yang digunakan agar PE dapat saling berkomunikasi seperti

shared bus dan *direct point – to – point*. Penggunaan kedua teknik tersebut memiliki kelebihan yaitu sederhana dan mudah dalam penerapannya. Tetapi seiring dengan meningkatnya ukuran dari jaringan akan menimbulkan permasalahan seperti *bottleneck* pada *shared bus* dan meningkatnya kompleksitas dan *delay* pada *direct point – to – point*. Salah satu solusi yang dapat digunakan untuk mengatasi permasalahan tersebut adalah *Network on Chip* (NoC). Dimana NoC dapat meningkatkan reliabilitas , efisiensi dan *delay* interkoneksi pada jaringan.

NoC memiliki parameter desain dan parameter performansi jaringan. Dimana penentuan parameter desain yang tidak tepat dan perubahan dari jaringan dapat menimbulkan permasalahan pada jaringan seperti *congestion* dan saturasi. *Congestion* terjadi karena adanya overload pada penerima dimana adanya ketidak sesuaian antara ukuran buffer dan ukuran paket yang dikirimkan serta tidak diketahuinya informasi mengenai buffer penerima oleh pengirim. *Congestion* dapat menyebabkan paket yang diterima di drop yang berakibat pada *error* dan menurunnya performa jaringan. Jaringan yang tersaturasi akan menyebabkan *throughput* pada jaringan mencapai nilai maksimum meskipun parameter desain lainnya diubah.

Congestion dan saturasi dapat diatasi dengan menerapkan *flow control* pada jaringan. *Flow control* digunakan untuk mencegah terjadinya antrian yang melebihi kapasitas, mengatur pengalokasian *buffer* dan titik pada setiap *router* dan menentukan kapan *buffer* dan link yang digunakan. Dalam memilih *flow control* harus diperhatikan pengaruh yang diberikan terhadap parameter desain seperti *overhead*, area yang digunakan dan pengkabelan dimana dapat mempengaruhi performansi jaringan.

Pada penelitian ini akan dilakukan analisis terhadap tiga buah teknik *flow control* yang ada pada NoC yaitu Stall / Go , Ack / Nack serta Dynamic Multi Level. Ketiga teknik *flow control* tersebut diterapkan pada dua model jaringan. Model jaringan yang pertama digunakan untuk mengamati pengaruh *flow control* terhadap saturasi jaringan. Sedangkan model jaringan yang kedua digunakan untuk mengamati pengaruh *flow control* terhadap perubahan parameter desain dan untuk mendapatkan teknik *flow control* yang paling optimal.

Adapun sistematika penulisan dalam paper ini adalah pada

bab 1 dijelaskan pendahuluan dan latar belakang. Bab 2 menjelaskan mengenai dasar teori yang berisi definisi – definisi yang digunakan dalam penelitian ini. Bab 3 akan dijelaskan mengenai mekanisme yang digunakan dalam mengatasi serangan pada jaringan. Bab 4 akan menampilkan hasil simulasi dan analisa performansi dari mekanisme yang digunakan. Bab 5 memberikan kesimpulan dari paper dan saran yang dapat dilakukan pada penelitian selanjutnya

II. DASAR TEORI

A. System on Chip

Pada SoC, beberapa sistem elektronik dan sistem komputasi yang disebut sebagai *Processing Element* (PE) ditanam pada sebuah chip. Teknologi yang digunakan pada komunikasi antar core yaitu *direct point – to – point*, *shared bus* dan *network on chip*.

Direct point – to – point merupakan teknik yang digunakan pertama kali pada SoC dimana setiap PE saling berkomunikasi satu sama lain melalui kabel yang menghubungkan setiap PE. *Direct point – to – point* memiliki skalabilitas yang rendah dan kompleksitas yang tinggi bila diterapkan pada jaringan yang besar. Namun untuk jaringan yang kecil, teknik ini memberikan performansi yang lebih baik.

Shared bus menggunakan *interface* untuk menghubungkan bus dengan setiap PE dan bus *arbiter* untuk mengatur proses komunikasi setiap PE. Teknik ini membutuhkan input dan output pin yang sedikit sehingga area untuk pengkabelan dan biaya yang diperlukan murah. Namun skalabilitas dari *shared bus* kurang baik dan proses pengiriman data lambat karena adanya *arbitration* dan jika terjadi *contention*.

Network on Chip memiliki infrastruktur yang fleksibel dan mengadopsi arsitektur dari jaringan komputer untuk mengirimkan dan menerima paket antar PE. NoC menggunakan router untuk meneruskan paket dari pengirim dan penerima. NoC menggunakan OSI model seperti jaringan komputer pada umumnya dimana tiga layer yang paling utama adalah layer fisik untuk sinyal clock dan sinyal kontrol, layer *data link* untuk *flitization*, *deflitization*, *error detection* dan *correction* dan layer *network* untuk *packetization*, *routing*, *buffering*, *congestion detection* and *control*.

B. Arsitektur Network on Chip

Arsitektur dari NoC direpresentasikan sebagai kombinasi antara *Processing Element* (PE), *Resource Network Interface* (RNI) dan juga router.

PE dapat berupa komponen hardware seperti processor, FPGA, Amplifier, DSP. Sedangkan RNI bertugas untuk menghubungkan PE dengan router sehingga setiap PE dapat mengirimkan pesan ke router tujuan. Router yang digunakan memiliki tugas utama untuk mentransmisikan pesan yang diterima PE ke tujuan apabila router terhubung secara langsung dengan PE tujuan atau meneruskan paket ke router tujuan. Router yang digunakan terdiri dari lima port yaitu Utara, Selatan, Barat, Timur yang terhubung dengan router

lainnya dan port lokal yang terhubung dengan PE. Setiap router memiliki buffer yang berfungsi sebagai tempat penyimpanan sementara.

C. Parameter Desain dan Performansi Network on Chip

Dalam mendesain sebuah jaringan pada NoC, parameter desain yang harus diperhatikan yaitu topologi, teknik switching, dan algoritma routing dengan permasalahan utama yaitu bagaimana dapat menghubungkan setiap PE hingga dapat saling berkomunikasi satu sama lain dengan performansi yang maksimal.

Topologi dapat didefinisikan sebagai layout fisik yang mendeskripsikan bagaimana router pada NoC dapat terhubung satu sama lain dengan syarat skalabilitas tinggi dan konsumsi daya rendah. Dua jenis topologi yang sering digunakan adalah mesh dan torus. Teknik *switching* menentukan kapan paket yang datang harus dilayani oleh router dan menentukan kapan paket tersebut dikirimkan oleh router ke router berikutnya. Terdapat tiga jenis teknik *switching* yang sering digunakan yaitu *store and forward*, *wormhole switching* dan *virtual cut through*. Sedangkan algoritma routing menentukan rute yang akan dilalui oleh paket yang dikirimkan untuk mencapai tujuan. Algoritma routing dapat memberikan pengaruh terhadap kompleksitas desain router dan luas penggunaan area serta mempengaruhi konsumsi daya.

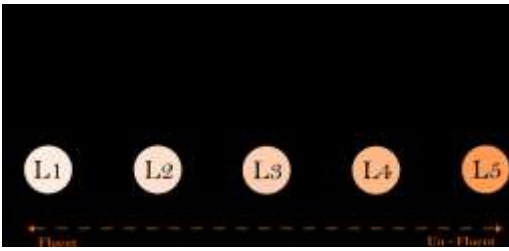
Parameter performansi pada NoC dapat berupa throughput, *delay* dan konsumsi daya pada jaringan. Dimana parameter performansi dipengaruhi oleh parameter desain yang digunakan.

Delay didefinisikan sebagai waktu antara saat PE sumber mengirimkan flit pertama ke jaringan dan saat PE tujuan menerima flit terakhir dari paket yang dikirimkan. *Delay* dihitung dalam satuan cycles. Sedangkan throughput dapat dikatakan sebagai laju pengiriman paket pada jaringan yang diterima oleh node tujuan dalam satuan waktu dan dinyatakan dengan satuan flits/cycle. Sedangkan penggunaan daya pada NoC dihitung dalam satuan Joule yang merupakan daya total yang digunakan oleh semua komponen dan proses yang terjadi pada jaringan.

D. Flow Control

Flow control mengatur pengalokasian *buffer* dan *link* pada tiap router dan menentukan kapan *buffer* dan *link* digunakan untuk suatu paket. Pemilihan *flow control* harus sesuai dengan kebutuhan karena dapat mempengaruhi kompleksitas pengkabelan, luas area, *overhead* yang mempengaruhi performansi. Beberapa *flow control* yang dapat digunakan pada NoC yaitu *Flow Control Stall / Go*, *Ack / Nack* dan *Dynamic Multi Level (DyML)*.

Flow Control Stall / Go memerlukan dua kabel untuk kontrol yaitu untuk *flagging data availability* yang berarah maju dan signalling kondisi *buffer (Stall / Go)* yang berarah mundur. Pada jaringan yang menerapkan *flow control* ini, setiap node mengetahui availabilitas dari slot kosong pada *buffer node downstream*. Saat slot kosong node tujuan mencapai batas off maka sinyal Stall akan dikirim ke node



Gambar II.1 Perindahan Level pada Dynamic Multi Level Flow Control

Tabel II.1 Tingkatan Buffer Fluidity dan Buffer Fill pada Dynamic Multi Level Flow Control

Level	Buffer Fluidity	Buffer Fill
L0	Empty (<i>buffer fill</i> = 0)	Buffer Fill = Buffer Size
L1	Fluent	Buffer Fill = Buffer Size - 1
L2	Middle Fluent	Buffer Fill > Buffer Size / 2
L3	Less Fluent	Buffer Fill > Buffer Size / 4
L4	Near Un – Fluent	Buffer Fill > Buffer Size / 8
L5	Un – Fluent	

upstream hingga mencapai node sumber dan sumber akan berhenti mengirimkan flit. Router pada node perantara saat menerima sinyal Stall akan menyimpan flit yang datang di *output register* dan *flow control register* kemudian mengirimkan Sinyal Stall ke *node upstream* berikutnya. Node tujuan memproses data yang ada di *buffer* dan saat slot kosong mencapai batas On maka node tujuan akan mengirimkan sinyal Go ke *node upstream* hingga mencapai node sumber. Node sumber yang menerima sinyal Go akan melanjutkan proses transmisi.

Pada *flow control* Ack / Nack, *node* sumber tidak mengetahui kondisi *buffer* pada *node downstream*, dimana *node* sumber akan mentransmisikan flit secara terus menerus ke *node* tujuan. Apabila *node* tujuan menerima flit maka akan mengirimkan sinyal Ack ke *node* sumber dan mengirimkan sinyal Nack apabila tidak menerima flit yang diharapkan. Apabila *node* sumber menerima Nack maka akan dilakukan pengiriman ulang.

Pada *flow control* DyML menggunakan konsep dasar dari *buffer fluidity* dimana terdapat enam kondisi yang digunakan dalam *flow monitor* untuk merepresentasikan kondisi jaringan sesuai dengan Gambar II.1. Dimana L0 hingga L5 merupakan tingkatan dari *buffer fluidity*, BF0 adalah kejadian saat *buffer fill* kosong sedangkan BF!0 saat *buffer* tidak kosong. NPTO adalah *Non Pop Time Out* dan PTO adalah *Pop Time Out* dimana *Pop* merupakan kejadian saat *buffer* meneruskan flit ke *node* berikutnya. Pada DyML, setiap tingkatan dipengaruhi oleh *buffer fluidity* dan *buffer fill* sesuai Tabel II.1

III. RANCANGAN JARINGAN DAN IMPLEMENTASI FLOW CONTROL

A. Model Jaringan Pertama

Pada model jaringan yang pertama digunakan untuk mengamati pengaruh *flow control* terhadap saturasi pada jaringan dengan parameter yang diamati titik saturasi dan *throughput* maksimum. Dimana saturasi pada jaringan terjadi

dengan menggunakan kombinasi parameter – parameter desain yang sesuai. Sehingga digunakan dua skenario pengujian yaitu yang pertama kombinasi antara ukuran *buffer* dengan *packet injection rate* yang berubah dan skenario kedua kombinasi *packet injection rate* dan ukuran paket yang berubah. Dimana parameter desain yang nilainya tetap yaitu waktu simulasi 100000 cycles, topologi 2D Mesh, ukuran jaringan 5 x 5, algoritma routing XY, *selection function random*, dan model trafik transpose. Sedangkan parameter desain yang berubah – ubah nilainya yaitu Ukuran *buffer* 2 – 10 flits, ukuran paket 2 – 22 flits, *packet injection rate* 0.02 – 0.3.

B. Model Jaringan Kedua

Model jaringan kedua digunakan untuk mengamati pengaruh perubahan parameter desain dan ketiga *flow control* terhadap parameter performansi. Dimana parameter performansi yang diamati adalah *throughput*, *delay* dan konsumsi daya. Pada model jaringan kedua terdapat 4 skenario yang digunakan. Skenario 1 adalah ukuran jaringan berubah – ubah nilainya dari 2 x 2 hingga 9 x 9. Skenario 2 adalah *packet injection rate* berubah – ubah dari 0.02 hingga 0.18. Skenario 3 adalah ukuran paket berubah – ubah dari 2 hingga 18 flit. Skenario 4 adalah ukuran *buffer* berubah – ubah dari 1 hingga 17 flit.

Adapun parameter desain untuk keseluruhan memiliki nilai tetap seperti waktu simulasi 100000 cycle, topologi 2D – Mesh, algoritma routing XY, *Selection function Random*, Model Trafik Transpose, *packet injection rate* 0.06, ukuran paket 2 – 4 flit, ukuran *buffer* 8 flit, ukuran jaringan 5 x 5.

C. Implementasi Flow Control

Setelah didapat algoritma untuk masing – masing metode *flow control* dan dengan model jaringan yang telah ada dapat dilakukan proses pengimplementasian algoritma kedalam Noxim sebagai simulator yang digunakan pada penelitian ini. Dikarenakan Noxim merupakan simulator yang menggunakan SystemC, maka algoritma yang sudah ada harus diubah ke dalam bentuk bahasa C++ terlebih dahulu dan diterapkan pada file – file tertentu pada Noxim. Untuk dapat mengimplementasikan *flow control* pada Noxim diperlukan adanya perubahan pada file *source* Noxim. Sehingga file – file *source* Noxim yang harus dirubah untuk dapat mengimplementasikan ketiga teknik *flow control* tersebut yaitu:

- NoximProcessingElement untuk mengatur pengiriman dan penerimaan flit di *Processing Element* tiap *router*
- NoximRouter untuk mengatur proses pengiriman dan penerimaan flit dan proses *routing* pada *router*
- NoximTile dan NoximNoC untuk pengaturan kanal arah maju dan mundur yang digunakan pada NoC
- NoximMain untuk melakukan pendefinisian parameter *global* yang digunakan
- NoximBuffer untuk mengatur bagaimana operasi – operasi pada *buffer*

IV. HASIL DAN ANALISA

A. Analisis Pengaruh Flow Control terhadap Saturasi di Jaringan

Dilakukan analisis dengan parameter yang diamati merupakan titik saturasi dan throughput maksimum.

Untuk skenario pertama yang menggunakan kombinasi ukuran *buffer* dan *packet injection rate* yang berubah – ubah hasil yang diperoleh sesuai dengan Tabel IV.1. Untuk jaringan yang hanya menggunakan *flow control default*, jaringan mengalami saturasi dikarenakan pengirim hanya dapat mengirimkan flit dengan *sliding window* 1 dan memerlukan *signalling* dari penerima sebelum dapat mengirimkan flit berikutnya. Sehingga dengan ukuran *buffer* dan ukuran flit yang tetap, sedangkan *packet injection rate* terus diperbesar akan dicapai kondisi dimana jumlah flit yang semakin banyak di jaringan tidak akan meningkatkan *throughput*. Sedangkan pada jaringan yang menerapkan *flow control* Stall / Go , Ack / Nack dan DyML tidak menyebabkan saturasi pada jaringan dikarenakan jaringan dapat mengirimkan flit secara terus menerus dan tidak terbatas oleh *sliding window*. Sehingga apabila jumlah paket yang dikirimkan semakin banyak sebagai akibat dari peningkatan throughput, tidak akan menyebabkan saturasi.

Untuk skenario kedua yang menggunakan kombinasi *packet injection rate* dan ukuran paket dengan mengacu pada hasil yang diperoleh sesuai Tabel IV.2 jaringan yang menggunakan *flow control* default mengalami saturasi. Sedangkan untuk *flow control* Stall / GO tidak terjadi saturasi dengan peningkatan ukuran paket hingga 20 – 22 flit. Dikarenakan flit dapat dikirimkan terus menerus selama *signalling* bertanda GO dan karena perubahan *signalling* menjadi STALL dipengaruhi oleh ukuran *buffer* yang mempengaruhi batas off dan batas on.

Tabel IV.1 Perbandingan Throughput Maksimum dan Titik Saturasi dengan kombinasi ukuran buffer dan packet injection rate yang berubah

No	Flow Control	Throughput Maksimum	Titik Saturasi (PIR)
1	Default	0.25742 flits / Cycle	0.18
2	Stall / Go	Tidak Tersaturasi	Tidak Tersaturasi
3	Ack / Nack	Tidak Tersaturasi	Tidak Tersaturasi
4	Dynamic Multi Level	Tidak Tersaturasi	Tidak Tersaturasi

Tabel IV.2 Perbandingan Throughput Maksimum dan Titik Saturasi dengan kombinasi packet injection rate dan ukuran flit yang berubah

No	Flow Control	Throughput Maksimum	Titik Saturasi (Ukuran Paket)
1	Default	0.25742 flits / Cycle	8 – 10 flit
2	Stall / Go	Tidak Tersaturasi	Tidak Tersaturasi
3	Ack / Nack	Tidak Tersaturasi	Tidak Tersaturasi
4	Dynamic Multi Level	Tidak Tersaturasi	Tidak Tersaturasi

Sedangkan pada flow control DyML dan Ack/Nack saat ukuran paket 8 – 10 flit terjadi throughput maksimum, namun saat ukuran paket diperbesar tidak terjadi saturasi melainkan throughput akan menurun. Hal tersebut disebabkan untuk jaringan dengan flow control Ack / Nack, flit dapat terus dikirimkan selama menerima Ack. Akan tetapi saat terjadi flit *drop* dan dikirimkan Nack harus dilakukan pengiriman ulang. Karena *buffer* yang digunakan hanya berukuran 8 flit saja maka saat ukuran paket yang digunakan lebih besar dari 8 – 10 flit kemungkinan untuk terjadinya flit *drop* dan pengiriman ulang semakin besar sehingga menyebabkan menurunnya throughput sesaat.

Sedangkan pada *flow control* DyML proses pengiriman flit bergantung pada kondisi *signalling* dan kondisi *buffer*. Dengan ukuran *buffer* yang tetap 8 flit kemungkinan *signalling* berada pada kondisi Stall untuk memproses flit yang terdapat di *buffer* akan semakin sering terjadi. Selain itu penentuan batas Off dan batas on serta rentang waktu pengecekan *Pop* juga menjadi faktor yang menyebabkan menurunnya throughput.

B. Analisis Pengaruh Parameter Desain dan Flow Control terhadap Performansi Jaringan

Pada model jaringan kedua dilakukan analisis pengaruh parameter desain dan *flow control* terhadap performansi jaringan berupa *throughput* , *delay* dan konsumsi daya dengan empat skenario berbeda. Hasil dari penelitian yang dilakukan untuk model kedua dapat dilihat pada Tabel IV.3, Tabel IV.4, dan Tabel IV.5.

Untuk skenario pertama dimana dengan ukuran jaringan yang berubah terhadap *throughput*, untuk jaringan dengan *flow control* default, *flow control* Ack/Nack dan *flow control* DyML, ukuran jaringan berbanding terbalik dengan throughput.. Sedangkan untuk *flow control* Stall / Go, dengan meningkatkan ukuran jaringan tidak menyebabkan perubahan throughput yang signifikan. Sehingga *flow control* Stall / Go memberikan peningkatan throughput terbesar yaitu 21.08% jika dibandingkan dengan *flow control* default.

Untuk skenario kedua dengan *packet injection rate* yang berubah terhadap throughput, untuk semua jaringan yang menggunakan teknik *flow control* memberikan peningkatan throughput apabila *packet injection rate* ditingkatkan. Dengan melakukan perbandingan peningkatan throughput rata – rata masing – masing *flow control* terhadap *flow control* default, *flow control* Stall / Go memberikan peningkatan throughput terbaik sebesar 65.33%.

Untuk skenario ketiga dengan ukuran paket yang berubah terhadap throughput, untuk semua jaringan yang menggunakan teknik *flow control* memberikan peningkatan throughput apabila ukuran paket ditingkatkan. Dengan melakukan perbandingan peningkatan throughput rata – rata masing – masing *flow control* terhadap *flow control* default, *flow control* Stall / Go memberikan peningkatan throughput terbaik sebesar 151%.

Untuk skenario keempat dengan ukuran *buffer* yang berubah terhadap throughput, untuk jaringan yang menggunakan *flow control* Ack / Nack dan DyML ukuran *buffer* berbanding lurus

Tabel IV.3 Peningkatan Throughput (%) terhadap flow control default

Flow Control	Parameter Desain			
	Ukuran Jaringan	PIR	Ukuran Paket	Ukuran Buffer
Ack / Nack	14.96	59.21	53.19	2.54
Stall / Go	21.08	65.33	151	13.37
DyML	14.96	60.33	53.61	6.36

Tabel IV.4 Penurunan Delay (cycles) terhadap flow control default

Flow Control	Parameter Desain			
	Ukuran Jaringan	PIR	Ukuran Paket	Ukuran Buffer
Ack / Nack	403.52	601.63	1617.63	317.19
Stall / Go	407.85	606.03	1631.95	322.59
DyML	403.59	601.74	1611.35	316.86

Tabel IV.5 Penurunan Konsumsi Daya (%) terhadap flow control default

Flow Control	Parameter Desain			
	Ukuran Jaringan	PIR	Ukuran Paket	Ukuran Buffer
Ack / Nack	23.54	7.79	42.71	20.69
Stall / Go	68.67	61.33	49.93	68.22
DyML	25.41	9.32	43.27	19.77

dengan throughput. Sedangkan untuk jaringan yang menggunakan *flow control* Stall / Go dan Default, ukuran *buffer* tidak memberikan pengaruh yang signifikan terhadap perubahan *throughput*. Dimana dengan melakukan perbandingan terhadap *flow control default*, *flow control* Stall / Go memberikan peningkatan *throughput* terbaik sebesar 13.37%.

Untuk skenario pertama dimana dengan ukuran jaringan yang berubah terhadap *delay*, untuk jaringan dengan *flow control default*, ukuran jaringan berbanding lurus dengan *delay*. Sedangkan untuk *flow control* Stall / Go, ukuran jaringan tidak memberikan pengaruh yang signifikan terhadap *delay*. Pada *flow control* Ack/ Nack dan DyML, peningkatan ukuran jaringan berbanding lurus dengan *delay* hanya sampai ukuran jaringan 5x5. Sehingga *flow control* Stall / Go memberikan penurunan *delay* terbesar yaitu 407.85 cycles jika dibandingkan dengan *flow control default*.

Untuk skenario kedua dengan *packet injection rate* yang berubah terhadap *delay*, jaringan yang menggunakan *flow control default* dan Stall/Go, *packet injection rate* berbanding lurus dengan *delay*. Sedangkan untuk *flow control* Ack/Nack dan DyML, *delay* yang terjadi mengalami penurunan hingga titik tertentu kemudian *delay* akan naik. Dengan melakukan perbandingan peningkatan throughput rata – rata masing – masing *flow control* terhadap *flow control default*, *flow control* Stall / Go memberikan penurunan *delay* terbaik sebesar 606.03 cycles.

Untuk skenario ketiga dengan ukuran paket yang berubah terhadap *delay*, untuk semua jaringan yang menggunakan teknik *flow control* memberikan penurunan *delay* apabila ukuran paket ditingkatkan. Dengan melakukan perbandingan

penurunan *delay* rata – rata masing – masing *flow control* terhadap *flow control default*, *flow control* Stall / Go memberikan peningkatan throughput terbaik sebesar 1631.95 cycles.

Untuk skenario keempat dengan ukuran *buffer* yang berubah terhadap *delay* untuk jaringan yang menerapkan *flow control default* dan *flow control* Stall perubahan ukuran *buffer* tidak memberikan pengaruh signifikan terhadap perubahan *delay*. Sedangkan untuk *flow control* Ack/Nack dan DyML ukuran *buffer* berbanding lurus dengan *delay*. Dimana dengan melakukan perbandingan terhadap *flow control default*, *flow control* Stall / Go memberikan penurunan *delay* terbaik sebesar 322.59 cycles

Untuk skenario pertama dimana dengan ukuran jaringan yang berubah terhadap konsumsi daya, untuk semua jaringan dengan *flow control* ukuran jaringan berbanding lurus dengan konsumsi daya. Sehingga *flow control* Stall / Go memberikan penurunan konsumsi daya terbesar yaitu 68.67% jika dibandingkan dengan *flow control default*.

Untuk skenario pertama dimana dengan ukuran jaringan yang berubah terhadap konsumsi daya, untuk semua jaringan dengan *flow control* ukuran jaringan berbanding lurus dengan konsumsi daya. Sehingga *flow control* Stall / Go memberikan penurunan konsumsi daya terbesar yaitu 61.33% jika dibandingkan dengan *flow control default*.

Untuk skenario ketiga dengan ukuran paket yang berubah terhadap konsumsi daya, untuk jaringan yang menggunakan *flow control default*, saat ukuran paket yang digunakan adalah 8 – 10 flit mulai terjadi saturasi pada konsumsi daya. Sedangkan untuk jaringan yang menggunakan *flow control* Stall / Go, Ack /Nack dan DyML konsumsi daya berbanding lurus dengan ukuran paket. Dimana dengan melakukan perbandingan terhadap *flow control default*, *flow control* Stall / Go memberikan penurunan konsumsi daya terbaik sebesar 49.93%.

Untuk skenario keempat dengan ukuran *buffer* yang berubah terhadap konsumsi daya, untuk jaringan yang menggunakan *flow control* Stall / Go dan *default*, peningkatan ukuran *buffer* tidak berdampak pada konsumsi daya. Sedangkan untuk *flow control* Ack/Nack dan DyML ukuran *buffer* berbanding lurus dengan konsumsi daya. Dimana dengan melakukan perbandingan terhadap *flow control default*, *flow control* Stall / Go memberikan penurunan konsumsi daya terbaik sebesar 68.22%.

V. KESIMPULAN/RINGKASAN

Kesimpulan yang dapat diambil dari penelitian ini yaitu Pada model jaringan pertama, *flow control* Stall / Go, Ack / Nack dan DyML tidak menyebabkan jaringan mengalami saturasi hingga *packet injection rate* yang digunakan sebesar 0.3 untuk kombinasi pertama dan hingga ukuran paket yang digunakan sebesar 20 – 22 flit untuk kombinasi kedua.

Pada model jaringan kedua, throughput dapat ditingkatkan dengan memperbesar *packet injection rate*, ukuran paket, ukuran *buffer* dan ukuran jaringan. *Delay* dapat diturunkan

dengan memperkecil *packet injection rate*, ukuran paket, ukuran jaringan dan memperbesar ukuran *buffer*. Penggunaan daya pada jaringan dapat dikurangi dengan memperkecil *packet injection rate*, ukuran paket, ukuran *buffer* dan ukuran jaringan.

Pada model jaringan kedua, teknik *flow control Stall / Go* memberikan hasil terbaik dalam peningkatan throughput rata – rata jika dibandingkan dengan *flow control default* sebesar 21.08% terhadap perubahan ukuran jaringan, 65.33% terhadap perubahan *packet injection rate*, 151% terhadap perubahan ukuran paket dan 13.37% terhadap perubahan ukuran *buffer*. teknik *flow control Stall / Go* memberikan hasil terbaik dalam penurunan *delay* rata – rata jika dibandingkan dengan *flow control default* sebesar 407.85 cycles terhadap perubahan ukuran jaringan, 606.03 cycles terhadap perubahan *packet injection rate*, 1631.95 cycles terhadap perubahan ukuran paket dan 322.59 cycles terhadap perubahan ukuran *buffer*. Selain itu teknik *flow control Stall / Go* memberikan hasil terbaik dalam penurunan penggunaan daya rata – rata jika dibandingkan dengan *flow control default* sebesar 68.67 % terhadap perubahan ukuran jaringan, 61.33% terhadap perubahan *packet injection rate*, 49.93% terhadap perubahan ukuran paket dan 68.22% terhadap perubahan ukuran *buffer*.

Saran untuk penelitian kedepannya dapat dilakukan perbandingan untuk kombinasi parameter desain yang lain untuk dapat mengetahui pengaruhnya terhadap kondisi saturasi pada jaringan. Dapat pula dilakukan implementasi model jaringan yang menggunakan *flow control* dan tidak menggunakan *flow control* pada perangkat hardware.

UCAPAN TERIMA KASIH

Penulis Muhammad Hizrian H mengucapkan terima kasih kepada Bapak Istas Pratomo dan Bapak Djoko Suprajitno atas Ilmu dan Bantuan yang telah diberikan untuk mengerjakan penelitian ini.

DAFTAR PUSTAKA

- [1] J. Howard, "A 48 – Core IA – 32 Processor in 45 – nm CMOS Using on – Die Message – Passing and DVFS for Performance and Power Scaling", IEEE Journal of Solid State Circuits, Vol. 46 no1, pp – 173 – 183, jan 2011
- [2] J. Kumar, A. Kanta, T. N. Kamal, K. K. Mahaptra, "Performance Evaluation of Different Routing Algorithm in Network on Chip", IEEE Asia Pacific Conference on Post Graduate Research in Microelectronics and Electronics, 2013
- [3] R. Marculescu, U. Y. Orgras, L. S. Peh, N. E. Jerger, Y. Hoskote, "Outstanding Research Problems in NoC Design : System Microarchitecture", IEEE Trans. Computer – aided Design of Integrated Circuits System, Vol. 28, No 1, pp 3 – 21 Jan, 2009
- [4] W. C. Tsai, Y. C. Lan, S. J. Chen, Y. Hen Hu, "DyML : Dynamic Multi Level Flow Control for Network on Chip", IEEE, 2010
- [5] Williams James Dally and Brian Towles, "Principles and Practices of Interconnection Networks", Morgan Kaufmaan, 2004
- [6] V. V. Nimbalkar, K. Varghese, "in – Channel Flow Control Scheme for Network on Chip", IEEE 13th Euromicro Conference on Digital System Design Architecture, Methods and Tools, 2010
- [7] J. Hu, R. Marculescu, "DyAD – Smart Routing for Networks on Chip" Proceedings 41st Design Automation Conference, 2004

- [8] W. Zhang, L. Hou, J. Wang, S. Geng, W. Wu, "Comparison Research between XY and Odd – Even Routing Algorithm of a 2 – Dimension 3 x 3 Mesh Topology Network on Chip", IEEE Global Congress on Intelligent Systems, 2009
- [9] XIAO Canwen, Z. Minxuan, D. Yong, Z. Zhitong, "Dimensional Bubble Flow Control and Fully Adaptive Routing in the 2 – D Mesh Network on Chip", IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2008
- [10] A. Pullini, F. Angielini, D. Bertozzi, L. Benini, "Fault Tolerance Overhead in Networks on Chip Flow Control Schemes", ACM, SBCCI, Brazil, September 4- 7
- [11] Y.C. Lan, M.C. Chen, A.P. Su, Y. H. Hu, S. J. Chen, "Fluidity Concept for NoC : A Congestion Avoidance and Relief Routing Scheme", IEEE 21st International SoC Conference, pp 65 – 70, 2008
- [12] M. Bakhouya, A. Chariete, J.Gaber, M. Wack, S. Niar, E. Coatanea, "Performance Evaluation of a Flow – Control Algorithm for Network on Chip", IEEE, 2012
- [13] I.Pratomo, "Adaptive NoC for Reconfigurable SoC"
- [14] C. Nicopoulos, V. Narayanan, C. R. Das, "Network on Chip Architectures : A Holistic Design Exploration", Springer, 2009
- [15] F. Gebali, H. Elmiligi, M. W. El – Kharashi, "Network on Chips : Theory and Practice", CRC Press, 2009